Muximus API Documentation

MuxLab Part # 500813 / API Document v1.6

INTRODUCTION

The Muximus API allows you to control your Muximus Network Controller for integration with third party tools.

This version of the API documentation applies to Muximus version 3.1.

CONNECTING TO THE API - HTTP METHOD

Connecting to the API is as simple as including the Bearer Token header in your HTTP requests. You can obtain your Bearer Token in Muximus by logging in, under Profile > Password > API Key. The header would look like this:

```
Authorization: Bearer <my-token-here>
```

Please note, each user in Muximus has a separate API key.

You can find relevant information about HTTP Authentication Headers over here: `https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication.`

The API is called via the unit's IP address, using port 5001 as follows:

```
https://[device-ip]:5001/[endpoint]
```

Unless specified otherwise, the `Content-Type` is `application/json`

The APIs are detailed in the sections that follow.

API Endpoints - HTTP METHOD

Presets: *A preset can include connections for RX-TX, Video Wall, Multiview, Signage, and Dante. A preset can specify the connection type (audio, video, usb, etc) as well as streaming protocols (multicast, RTSP etc).*

```
Method: POST

Path: /presets/{PRESET_ID}/activate
```

Description: Activate a Preset that was created in the "Presets & Routing" page. The preset ID can be found in the 3-dots (Kebab) ⋮ menu of a saved preset.

Controls: *A control can include a command for RS-232, IR, and CEC.*

```
Method: POST

Path: /devices/ send-data-rs232-by-device-id

Body: {

    "device_id": "DEVICE-ID",

    "hexdata": "hexadecimal data"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
Method: POST

Path: /devices/ send-data-ir-by-device-id

Body: {

    "device_id": "DEVICE-ID",

    "hexdata": "hexadecimal data"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
Method: POST
```

```
Path: /devices/ send-cec-command-by-device-id

Body: {

    "device_id": "DEVICE-ID",

    "command": "powerOn"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⠿ menu in the Device Manager page.

`command` can be any of these values: 'standBy', 'powerOn', 'volumeUp', 'volumeDown', 'volumeMute'.

Connections: *Connect receivers, transmitters and multiviews.*

For the commands below that specify a Transmitter ID, you can use `00-00-00-00-00-00` to STOP the video playback.

Transmitter to Receiver

```
Method: POST

Path: /devices/connect

Body: {

    "presetConnections": [

        {

            "receiver_uuid": "RX-ID",

            "transmitter_uuid": "TX-ID",

            "connectionType": "all"

        }

    ]

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⠿ menu in the Device Manager page.

`connectionType` can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'

`protocol` is optional, 'auto' by default and can be any of these values, according to what the device in question supports: 'auto', 'rtsp', 'hls', 'ts', 'flv', 'multicast'

```
Method: POST

Path: /devices/connect-direct

Body: {

    "receiver_uuid": "RX-ID",

    "transmitter_uuid": "TX-ID"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

`connectionType` is optional, 'all' by default and can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'.

`protocol` is optional, 'auto' by default and can be any of these values, according to what the device in question supports: 'auto', 'rtsp', 'hls', 'ts', 'flv', 'multicast'

Multiview to Receiver

Important: This API will be deprecated in muximus version 4.

```
Method: POST

Path: /multiviews/connect

Body: {

    "multiviewPresetConnections": [

        {

            "multiview_uuid": "MULTIVIEW-ID",

            "receiver_uuid": "RX-ID",

            "receiver_type": "rx"

        }

    ]
```

```
}
```

500890 Signage Commands: Send AV over IP content to your 500890 Signage player, either to the whole screen, or to a window (zone) on the screen.

Using a transmitter as a source

```
Method: POST

Path: /signage/connect

Body: {

    "presetConnections": [

        {

            "receiver_uuid": "RX-ID (500890 model)",

            "transmitter_uuid": "TX-ID",

            "window_id": "all", or "1" to "6",

            "protocol": "auto", or "RTSP", "TS", "MULTICAST"

        }

    ]

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⠿ menu in the Device Manager page.

Using a multiview as a source, full screen only (cannot specify a window).

Important: This API will be deprecated in muximus version 4.

```
Method: POST

Path: /multiviews/connect

Body: {

    "multiviewPresetConnections": [

        {
```

```
            "multiview_uuid": "MULTIVIEW-ID",

            "receiver_uuid": "RX-ID (500890 model)",

            "receiver_type": "signage"

        }

    ]

}
```
Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

Reboot: *reboot a device.*

```
Method: POST

Path: /devices/ reboot-by-device-id

Body: {

    "device_id": "DEVICE-ID",

}
```
Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

Blink: *Triggers the front LED to blink. Supported on all MIP 600 devices except model 500890.*

```
Method: POST

Path: /devices/ blink-by-device-id

Body: {

    "device_id": "DEVICE-ID",

}
```
Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

CONNECTING TO THE API - TCP or UDP METHOD

Connecting to the API is as simple as sending a TCP or UDP command.

The TCP port is **9000**.

The UDP port is **9500**.

The command would look like this:

```
IP: Unit IP
Port: 9000 (TCP) or 9500 (UDP)
Command: command <arg1>, ...
```

The APIs are detailed in the sections that follow.

API - TCP or UDP METHOD

Presets: *A preset can include connections for RX-TX, Video Wall, Multiview, Signage, and Dante. A preset can specify the connection type (audio, video, usb, etc) as well as streaming protocols (multicast, RTSP etc).*

```
ACTIVATE_PRESET <presetId>
```

Description: Activate a Preset that was created in the "Presets & Routing" page. The preset ID can be found in the 3-dots (Kebab) ⋮ menu of a saved preset.

Controls: *A control can include a command for RS-232, IR, and CEC.*

SEND_RS-232_COMMAND <deviceId> <hexadecimal data>

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

SEND_IR_COMMAND <deviceId> <hexadecimal data>

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

SEND_CEC_COMMAND <deviceId> <command>

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`command` can be any of these values: 'standBy', 'powerOn', 'volumeUp', 'volumeDown', 'volumeMute'.

Connections: *Connect receivers, transmitters.*

For the commands below that specify a Transmitter ID, you can use `00-00-00-00-00-00` to STOP the video playback.

Transmitter to Receiver

CONNECT_RECEIVER <receiverId> <transmitterId>

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

500890 Signage Commands: Send AV over IP content to your 500890 Signage player, either to the whole screen, or to a window (zone) on the screen.

CONNECT_SIGNAGE <signageId> < transmitterId > <windowId>

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

window_id: "all", or "1" to "6". Not required, 'all' by default.

Reboot: *reboot a device.*

REBOOT_DEVICE <deviceId>

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

Blink: *Triggers the front LED to blink. Supported on all MIP 600 devices except model 500890.*

BLINK_DEVICE <deviceId>

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

SUPPORT

For support in integrating these APIs, please send a request to support@muxlab.com and you will put in touch with someone who can assist you with your API integrations.